

FIG.1

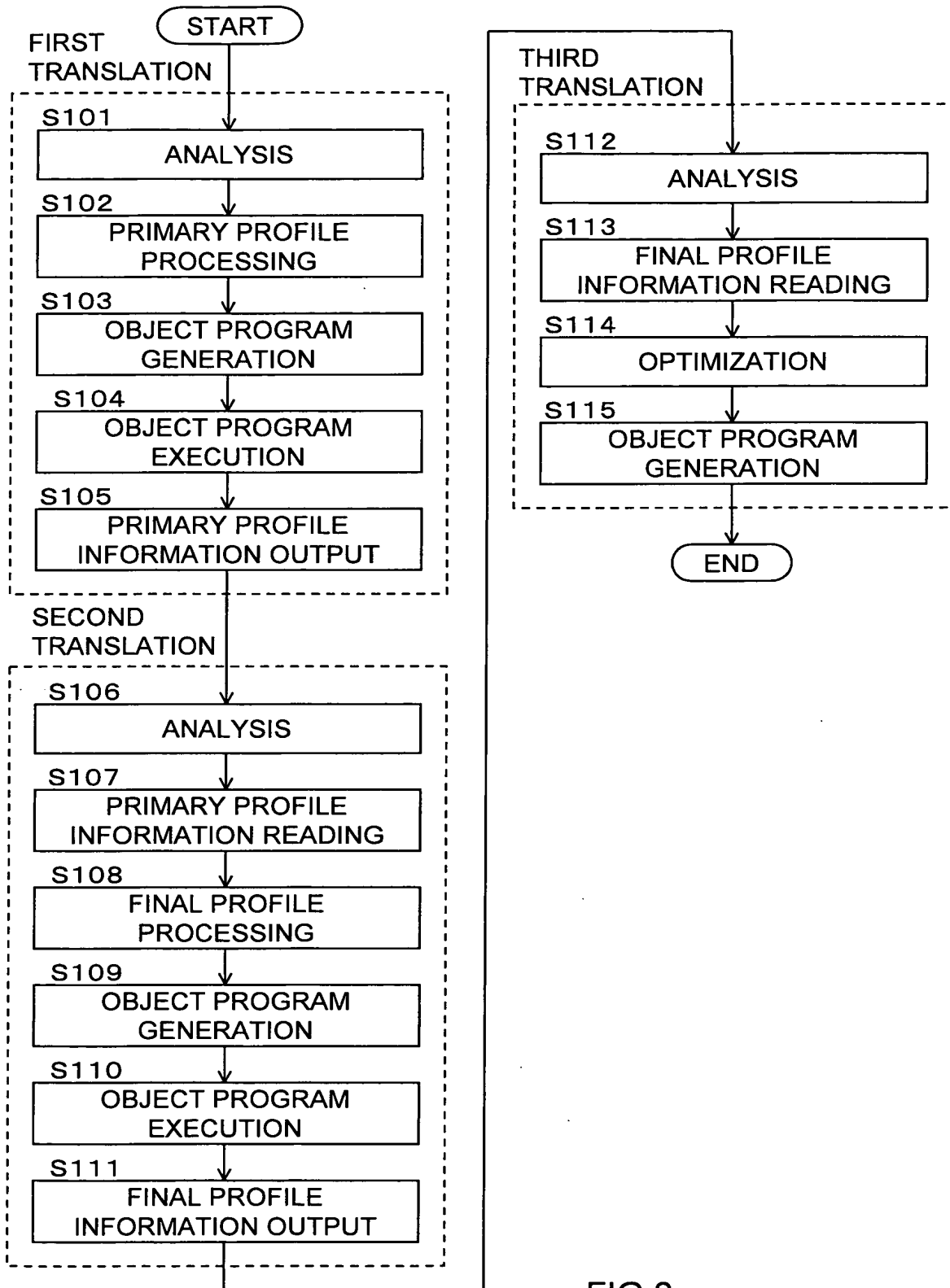


FIG.2

6 ~

BASIC BLOCK NUMBER	
BASIC BLOCK EXECUTION COUNTER	
VARIABLES NUMBER	
VALUE LIST A	VALUE Va
	COUNTER Na
VALUE LIST B	VALUE Vb
	COUNTER Nb

FIG.3A

```

struct prof_record {
    int bb_id;          /* UNIQUE NUMBER ASSIGNED BY
                        /* COMPILER TO BASIC BLOCK */
    int bb_count;       /* COUNTER OF BASIC BLOCK EXECUTIONS */
    int var_id;         /* UNIQUE NUMBER ASSIGNED BY COMPILER
                        /* TO VARIABLE */
    struct {            /* VALUE LIST */
        int val;        /* VALUE ASSIGNED TO VARIABLE UPON
                        /* EXECUTION */
        int count;      /* NUMBER OF TIMES VALUE IS GIVEN */
    } array [2]        /* NUMBER OF VALUE LIST IS 2 */
}

```

FIG.3B

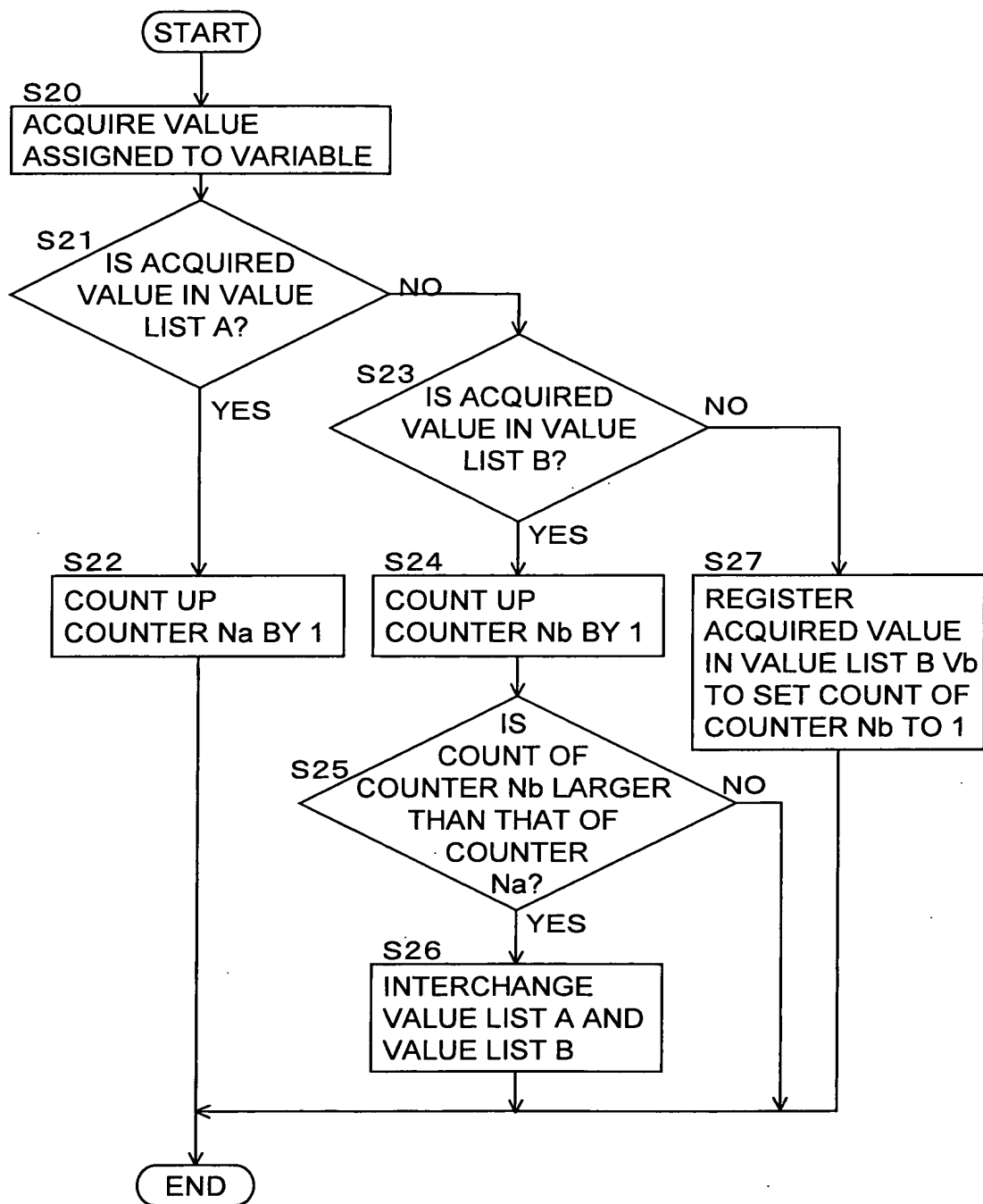


FIG.4

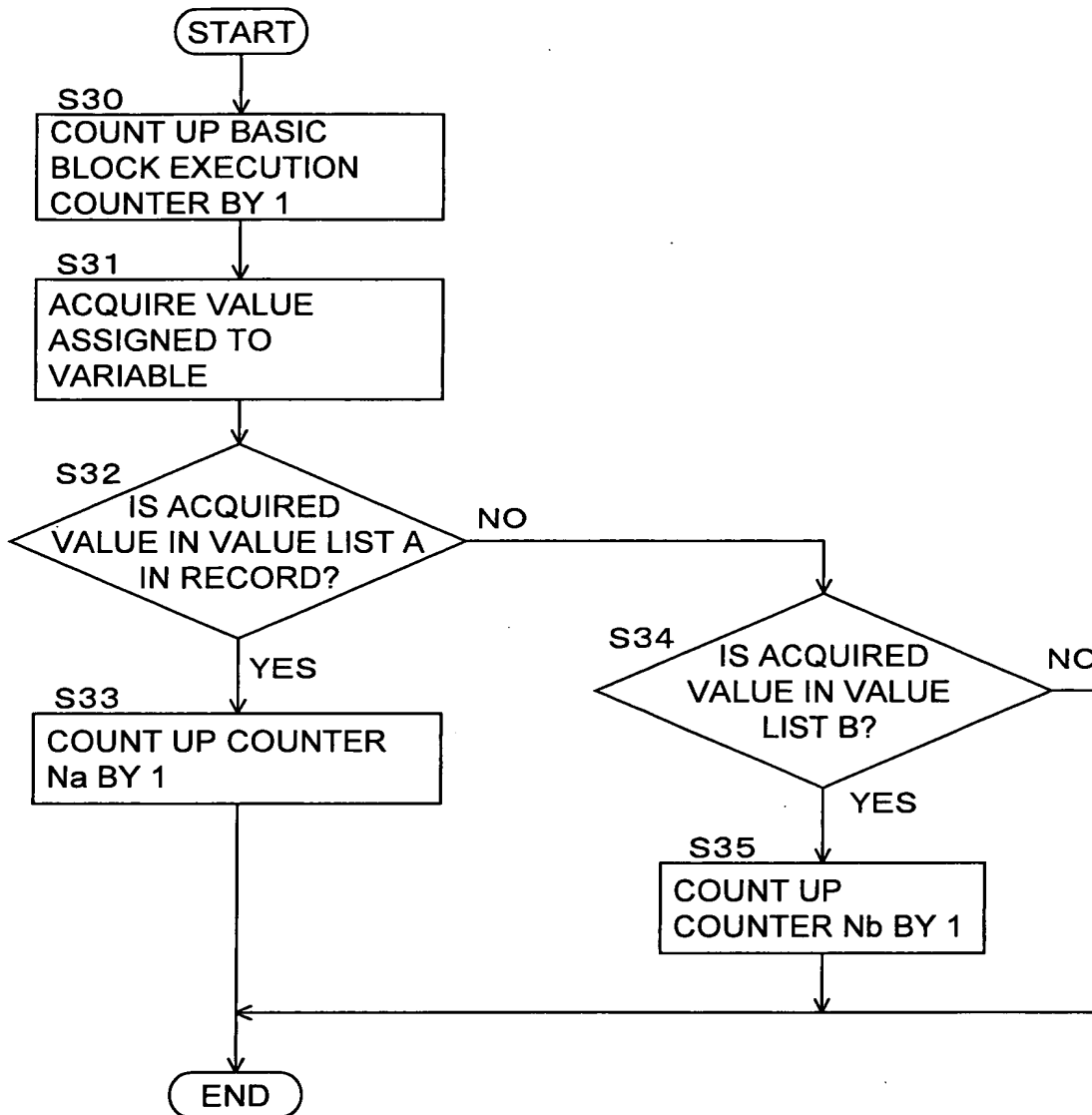


FIG.5

BEFORE OPTIMIZATION

```
short foo (short x, short y)
{
    short d;

    d = x / y;
    return d;
}
```

FIG.6A

AFTER OPTIMIZATION

```
short foo (short x, short y)
{
    short d;

    if (y == 17) {
        return ((int)x * 3855) >> 16;
    } else {
        d = x / y;
        return d;
    }
}
```

FIG.6B